

[First Hit](#) [Previous Doc](#) [Next Doc](#) [Go to Doc#](#)☐ [Generate Collection](#) [Print](#)

L8: Entry 13 of 21

File: JPAB

Jun 11, 1993

PUB-NO: JP405143373A
DOCUMENT-IDENTIFIER: JP 05143373 A
TITLE: SHARED DATA CONTROL SYSTEM

PUBN-DATE: June 11, 1993

INVENTOR-INFORMATION:

NAME

COUNTRY

REKIMOTO, JIYUNICHI

ASSIGNEE-INFORMATION:

NAME

COUNTRY

NEC CORP

APPL-NO: JP03329601

APPL-DATE: November 18, 1991

INT-CL (IPC): G06F 9/46; G06F 12/00; G06F 13/00

ABSTRACT:

PURPOSE: To prevent the generation of a contradiction in shared data without requiring the specified acquisition and release of operation right and to prevent the quantity of communication from being increased due to the guarantee of non-contradiction.

CONSTITUTION: A server process 4 is connected to plural client processes 1 to 3 through inter-process communication lines 5 to 7 and stores a time stamp 14 relating to shared data. When one (e.g. the client process 1) of the client processes 1 to 3 outputs a change request, the server process 4 returns permission information to the process 1 only when a time stamp 11 in the process 1 coincides with the time stamp 14. In addition, the process 4 reflects the change of data in one client process (e.g. the client process 1) to data 9, 10 in other client processes 2, 3.

COPYRIGHT: (C)1993,JPO&Japio

[Previous Doc](#) [Next Doc](#) [Go to Doc#](#)

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-143373

(43)公開日 平成 5 年(1993) 6 月11日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46	3 4 0 F	8120-5B		
12/00	5 3 5 Z	7832-5B		
13/00	3 5 5	7368-5B		

審査請求 未請求 請求項の数 1 (全 10 頁)

(21)出願番号 特願平3-329601

(22)出願日 平成 3 年(1991)11月18日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

(72)発明者 ▲暦▼本 純一

東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

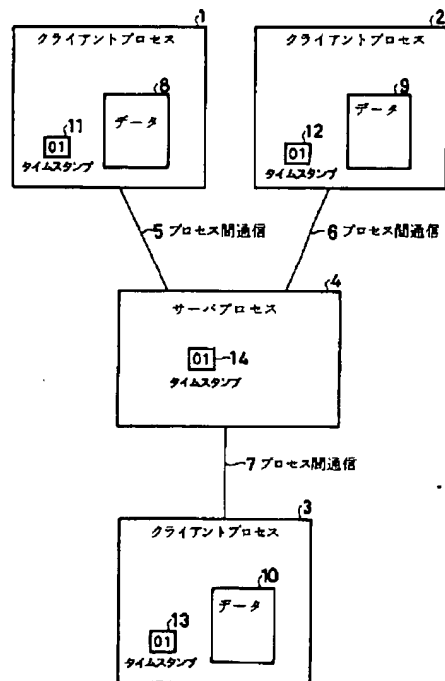
(74)代理人 弁理士 河原 純一

(54)【発明の名称】 共有データ制御方式

(57)【要約】

【目的】 操作権の明示的な取得および解放を要することなく、共有データに矛盾が生じないようにし、しかも非矛盾性の保証のための通信量の増大が生じないようにする。

【構成】 サーバプロセス4は、複数のクライアントプロセス1～3とプロセス間通信5～7によって結合され、共有データに関するタイムスタンプ14を保持している。サーバプロセス4は、クライアントプロセス1～3の中の1つ（クライアントプロセス1とする）からの変更要求に対し、クライアントプロセス1内のタイムスタンプ11とタイムスタンプ14とが同一の場合にのみクライアントプロセス1に許可通知を返送する。また、サーバプロセス4は、1つのクライアントプロセス（クライアントプロセス1とする）内のデータ8の変更を他のクライアントプロセス2および3内のデータ9および10に反映させる。



【特許請求の範囲】

【請求項1】 共有データのコピーのデータとそのデータに関するタイムスタンプとを保持するクライアントプロセスと、

複数の前記クライアントプロセスとプロセス間通信によって結合され、当該共有データに関するタイムスタンプを保持し、複数の前記クライアントプロセスの中の1つの前記クライアントプロセスからの変更要求に対して当該1つの前記クライアントプロセス内のタイムスタンプと自己の保持するタイムスタンプとが同一の場合にのみ当該1つの前記クライアントプロセスに許可通知を発行し、複数の前記クライアントプロセスの中の1つの前記クライアントプロセス内のデータの変更を当該1つの前記クライアントプロセス以外の前記クライアントプロセス内のデータに反映させるサーバプロセスとを有することを特徴とする共有データ制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、計算機システムにおけるデータ制御方式に関し、特に複数の異なるクライアントプロセス（計算機システムの各利用者に対応しデータの保持を担当するプロセス）により共有されるデータ（共有データ）の制御を行う共有データ制御方式に関する。

【0002】

【従来の技術】次のような状況における共有データ制御方式について考える。

㊶ 複数のクライアントプロセスのそれぞれは、共有データ（データベースシステム等において管理されている共有データ）のコピーのデータを保持している。

㊷ あるクライアントプロセス内のデータの変更（非同期的な変更）はそのクライアントプロセス以外のクライアントプロセス群に実時間で通知され、全てのクライアントプロセス内のデータ（共有データのコピーのデータ）の内容が常に一致するように保たれる。

㊸ 複数のクライアントプロセスによる共有データの変更の要求に対して、1つのクライアントプロセスにのみ変更が許可され、複数のクライアントプロセス内のデータ（共有データのコピーのデータ）の間に矛盾が生じることを防ぐための排他制御が行われる。

【0003】このような状況は、例えば、計算機ネットワークが利用された電子会議システムを実現する際に現出する。すなわち、電子会議システムによって行われる会議の複数の参加者に対応する複数のクライアントプロセスは、それぞれ文書を示すデータ（共有データのコピーのデータ）を保持している。そして、どの参加者が文書の内容を変更してもその変更が他の全ての参加者に対応するクライアントプロセス内のデータに実時間で反映されなければならない。

【0004】このような電子会議システムでは、全ての

クライアントプロセス内の文書を示すデータは常に同じ内容になるように管理されることになり、会議の全ての参加者に「自分が他の参加者と同じ文書を共有しており他の参加者と共同でその文書を変更している」という感覚を与えることができる。

【0005】上述のような状況で共有データに関する制御を行う従来の共有データ制御方式では、ある時点においては複数のクライアントプロセスの中の1つのクライアントプロセスだけが共有データの変更の権利を持ち（この権利を「操作権」という）、操作権を複数のクライアントプロセスの間で交換することで各クライアントプロセスが保持するデータの間に不一致（矛盾）が生じることを防ぐという方式が採用されていた（このような方式を「操作権方式」ということにする）。

【0006】なお、共有データの制御に関する他の方式としては、各クライアントプロセスが保持するコピーのデータの間に矛盾が生じないように、「最初にコピーのデータを保持したクライアントプロセスがそのデータを共有データに書き戻すまで次のクライアントプロセスがその共有データのコピーのデータを取得できないようにする」という排他制御が行われる方式が存在する（このような方式を「ロック方式」ということにする）。

【0007】

【発明が解決しようとする課題】上述した従来の共有データ制御方式等（共有データに関する制御が行われない方式およびロック方式を含む）には、次のような問題点があった。

【0008】㊶ 共有データに関する一切の制御を行わない場合の問題点

30 この場合には、複数のクライアントプロセスが同時に同一の共有データに対して異なる変更を加えたときに、共有データの間に矛盾が生じる可能性がある。例えば、あるクライアントプロセスAが共有データの一部を削除する変更を行おうとしており、別のクライアントプロセスBがクライアントプロセスAによって削除されようとしている共有データの内容を別の内容に変更しようとしているとする。このときに、クライアントプロセスAからの削除の要求が先に処理されると、クライアントプロセスBからの変更の要求に対してはすでに存在しなくなった共有データの一部に関する変更が行われることになり、結果として矛盾した共有データが生じることになる。

【0009】㊷ 操作権方式（従来の共有データ制御方式）による場合の問題点

この場合には、上述の㊶の場合のような矛盾は生じない。なぜならば、ある瞬間に共有データのコピーのデータを変更することができるのは、1つのクライアントプロセスに限られているからである。しかし、操作権方式による場合には、次のような別の問題点がある。すなわち、最初に操作権を取得したクライアントプロセスがデ

ータ（共有データのコピーのデータ）を変更した後も操作権を持ち続けていると他のクライアントプロセスはいつまでも自己のデータ（共有データのコピーのデータ）を変更することができないので、操作権はできるだけ速やかに1つのクライアントプロセスから解放されるべきであるという要請がある。一方、1つの変更の要求毎に毎回操作権の取得および解放が行われるのでは、データの変更のための通信の他に操作権の制御のための余計な通信が必要になる（これによりこの方式が採用される計算機システムにおける処理の効率が低下してしまう）ので、操作権の取得および解放は少ない方が望ましいという要請がある。操作権方式（従来の共有データ制御方式）では、このような2つの相反する要請を適切に満たして、「いつ操作権を解放すべきか」「あるクライアントプロセスにいつまで操作権を保持させるべきか」という困難な決定が必要になるという問題点がある。

【0010】㊟ ロック方式による場合の問題点

この場合には、複数の利用者が実時間で共有データのコピーのデータを変更するためには、共有データのコピーおよび共有データへの書戻しを頻繁に繰り返す必要が生じ、この方式が採用される計算機システムの処理の効率が悪くなる。逆にいうと、ロック方式は複数のクライアントプロセスが実時間で共有データのコピーのデータを変更しあうような計算機システム（電子会議システム等）の応用には適していないといえる。

【0011】本発明の目的は、上述の点に鑑み、共有データに矛盾が起きるような変更要求を確実に拒否して複数のクライアントプロセス内のデータ（共有データのコピーのデータ）に矛盾が生じないことを保証することができ（しかも、その非矛盾性の保証のための余計な通信を不要にする）、操作権の明示的な取得および解放の必要がなくなる（「あるクライアントプロセスにいつまで操作権を保持させるべきか」の決定が不要になる）共有データ制御方式を提供することにある。

【0012】

【課題を解決するための手段】本発明の共有データ制御方式は、共有データのコピーのデータとそのデータに関するタイムスタンプとを保持するクライアントプロセスと、複数の前記クライアントプロセスとプロセス間通信によって結合され当該共有データに関するタイムスタンプを保持し複数の前記クライアントプロセスの中の1つの前記クライアントプロセスからの変更要求に対して当該1つの前記クライアントプロセス内のタイムスタンプと自己の保持するタイムスタンプとが同一の場合にのみ当該1つの前記クライアントプロセスに許可通知を発行し複数の前記クライアントプロセスの中の1つの前記クライアントプロセス内のデータの変更を当該1つの前記クライアントプロセス以外の前記クライアントプロセス内のデータに反映させるサーバプロセスとを有する。

【0013】

【作用】本発明の共有データ制御方式では、クライアントプロセスが共有データのコピーのデータとそのデータに関するタイムスタンプとを保持し、複数のクライアントプロセスとプロセス間通信によって結合されており当該共有データに関するタイムスタンプを保持しているサーバプロセスが複数のクライアントプロセスの中の1つのクライアントプロセスからの変更要求に対して当該1つのクライアントプロセス内のタイムスタンプと自己の保持するタイムスタンプとが同一の場合にのみ当該1つのクライアントプロセスに許可通知を発行し複数のクライアントプロセスの中の1つのクライアントプロセス内のデータの変更を当該1つのクライアントプロセス以外のクライアントプロセス内のデータに反映させる。

【0014】

【実施例】次に、本発明について図面を参照して詳細に説明する。

【0015】図1は、本発明の共有データ制御方式の一実施例の構成を示すブロック図である。本実施例の共有データ制御方式は、クライアントプロセス1～3と、プロセス間通信5～7によってクライアントプロセス1～3と通信を行うサーバプロセス4と、プロセス間通信5～7とを含んで構成されている。

【0016】クライアントプロセス1は、データ8と、データ8に関するタイムスタンプ11とを保持している。

【0017】クライアントプロセス2は、データ9と、データ9に関するタイムスタンプ12とを保持している。

【0018】クライアントプロセス3は、データ10と、データ10に関するタイムスタンプ13とを保持している。

【0019】サーバプロセス4は、データ8～10に係る共有データ（データ8～10はその共有データのコピーのデータに該当する）に関するタイムスタンプ14を保持している。

【0020】図2は、クライアントプロセス1～3がサーバプロセス4に変更要求を発行する際の処理を示す流れ図である。この処理は、変更要求等送信ステップ21と、返事待機ステップ22と、返事判定ステップ23と、タイムスタンプセットステップ24と、変更要求データ反映ステップ25とからなる。

【0021】図3は、クライアントプロセス1～3がサーバプロセス4からの変更通知を受け付ける際の処理を示す流れ図である。この処理は、変更通知受け付けステップ31と、タイムスタンプセットステップ32と、変更通知データ反映ステップ33とからなる。

【0022】図4は、サーバプロセス4がクライアントプロセス1～3の変更要求を受け付ける際の処理を示す流れ図である。この処理は、変更要求受け付けステップ41と、タイムスタンプ判定ステップ42と、タイムスタ

5

ンプ増加ステップ43と、許可通知等返送ステップ44と、変更要求等通知ステップ45と、不許可通知返送ステップ46とからなる。

【0023】図5は、本実施例の共有データ制御方式の具体的な動作を説明するためのブロック図(複数のクライアントプロセス(クライアントプロセス1および3)が同時に変更要求を発行する場合の本実施例の共有データ制御方式の動作を説明するためのブロック図)である。

【0024】次に、このように構成された本実施例の共有データ制御方式の動作について説明する。なお、以下の説明では、プロセス間通信5〜7については言及しないが、クライアントプロセス1〜3とサーバプロセス4との間で通信が行われる場合にはプロセス間通信5〜7が介在している。

【0025】各クライアントプロセス1〜3は、共有データのコピーであるデータ8〜10と、データ8〜10に関するタイムスタンプ11〜13とを保持している。

【0026】サーバプロセス4は、データを保持せずに、データ8〜10のコピー元である共有データに関するタイムスタンプ14を保持している。

【0027】初期状態では、全てのクライアントプロセス1〜3が保持しているデータ8〜10の内容は一致しており、全てのクライアントプロセス1〜3の持つタイムスタンプ11〜13の全てとサーバプロセス4の持つタイムスタンプ14とは一致している。

【0028】あるクライアントプロセス(クライアントプロセス1とする)でデータ8の変更に対する要求が発生すると、クライアントプロセス1はその変更の内容を示す変更要求とクライアントプロセス1内のタイムスタンプ11とを組にしてサーバプロセス4に送信し(ステップ21)、サーバプロセス4からの返事を待つ(ステップ22)。

【0029】サーバプロセス4は、クライアントプロセス1からの変更要求を受け付け(ステップ41)、サーバプロセス4内のタイムスタンプ14とステップ41で受け付けた変更要求と組になっているタイムスタンプ(クライアントプロセス1内のタイムスタンプ11)とを比較する(ステップ42)。

【0030】ステップ42の比較で両者が一致すれば、サーバプロセス4は、タイムスタンプ14を1だけ増加し(ステップ43)、変更要求を送信してきたクライアントプロセス1に許可通知(「許可」を示すフラグ)とタイムスタンプ14とを組にして返送し(ステップ44)、ステップ41で受け付けた変更要求とサーバプロセス4内のタイムスタンプ14とを組にして他のクライアントプロセス2および3に通知する(ステップ45)。

【0031】一方、ステップ42の比較で両者が一致しなければ、サーバプロセス4は、不許可通知(「不許

6

可」を示すフラグ)をクライアントプロセス1に返送する(ステップ46)。

【0032】クライアントプロセス2および3は、サーバプロセス4からの変更通知を受け付け(ステップ31)、その変更通知と組になっているタイムスタンプ(サーバプロセス4内のタイムスタンプ14)をクライアントプロセス2および3内のタイムスタンプ12および13としてセットし(ステップ32)、クライアントプロセス2および3が保持しているデータ9および10をその変更通知を反映して更新する(ステップ33)。

【0033】変更要求を送信したクライアントプロセス1は、サーバプロセス4からの返事を受け付けると、その返事が許可通知であるか否かを判定する(ステップ23)。

【0034】ステップ23の判定で返事が許可通知である場合には、クライアントプロセス1は、その許可通知と組になっているタイムスタンプ(サーバプロセス4内のタイムスタンプ14)をクライアントプロセス1内のタイムスタンプ11としてセットし(ステップ24)、ステップ21で送信した変更要求を反映して自己のデータ8の更新を行う(データ8の変更は成功する)(ステップ25)。

【0035】一方、ステップ23の判定で返事が許可通知でない場合(不許可通知である場合)には、クライアントプロセス1は、データ8の変更を行わない(データ8の変更は失敗する)。

【0036】次に、図5を参照して、2つのクライアントプロセス(クライアントプロセス1および3とする)が同時にデータ8および10の変更要求(後述する変更要求52および53)を発行した場合の本実施例の共有データ制御方式の動作について説明する。

【0037】まず、サーバプロセス4が変更通知51とタイムスタンプ14とを組にして各クライアントプロセス1〜3に送信している状況を想定する(図5中ではサーバプロセス4からクライアントプロセス1への変更通知51だけを示しているがクライアントプロセス2および3に対しても同一の変更通知が送信されている)。

【0038】変更通知51に関する図3に示す処理をまだ終了していないクライアントプロセス1は、変更要求52とクライアントプロセス1内のタイムスタンプ11(「02」を示している)とを組にしてサーバプロセス4に送信する。

【0039】同時に、サーバプロセス4からの変更通知(変更通知51と同一の変更通知)に関する図3に示す処理を終了したクライアントプロセス3も、別の変更要求53とクライアントプロセス3内のタイムスタンプ13(「03」を示している)とを組にしてサーバプロセス4に送信する。

【0040】サーバプロセス4は、2つの変更要求52および53を受け付け、両方の変更要求52および53

と組になっているタイムスタンプ(タイムスタンプ11および13)とサーバプロセス4内のタイムスタンプ14(「03」を示している)とを比較する。

【0041】この比較の結果(タイムスタンプ11とタイムスタンプ14とは一致せずに、タイムスタンプ13とタイムスタンプ14とは一致するとう結果)に基づいて、サーバプロセス4は、変更要求52に対しては不許可通知をクライアントプロセス1に返送し、変更要求53に対しては許可通知をクライアントプロセス3に返送する。したがって、クライアントプロセス1内のデータ8の変更は失敗し、クライアントプロセス3内のデータ10の変更は成功する。

【0042】次に、本実施例、ひいては本発明の共有データ制御方式と従来の技術である「操作権方式」とが併用される場合の動作について説明する。

【0043】この場合には、クライアントプロセス1〜3およびサーバプロセス4は、以下のようにして動作することになる。

【0044】① 操作権を取得したいクライアントプロセス1、2または3は、操作権の取得要求と自己が保持するタイムスタンプ11、12または13とを組にしてサーバプロセス4に送信する。

【0045】② サーバプロセス4は、クライアントプロセス1、2または3から送られてきたタイムスタンプ11、12または13と自己が保持するタイムスタンプ14とを比較する。

【0046】③ ②の比較で両者が一致する場合には、サーバプロセス4は、サーバプロセス4内のタイムスタンプ14を1だけ増加し、増加後のタイムスタンプ14と許可通知とを操作権の取得要求を送信してきたクライアントプロセス1、2または3に返送する。また、サーバプロセス4は、操作権の保持者としてそのクライアントプロセス1、2または3の識別情報を記録しておく。この状態を、「操作権が設定された状態」と呼ぶ。

【0047】④ ②の比較で両者が一致しない場合には、サーバプロセス4は、不許可通知を操作権の取得要求を送信してきたクライアントプロセス1、2または3に返送する。

【0048】⑤ 操作権が設定された状態では、操作権を保持するクライアントプロセス1、2または3からの変更要求以外に対しては、たとえその変更要求と組になっているタイムスタンプ11、12または13がサーバプロセス4内のタイムスタンプ14と一致したとしても、サーバプロセス4は全て不許可通知を返送する。

【0049】なお、本実施例では共有データが1つ(共有データのコピーのデータ8〜10が1組)である場合について述べたが、共有データが複数である場合にも、各共有データに対して別組のタイムスタンプを用意することによって、本発明を適用することができることはいうまでもない。

【0050】

【発明の効果】以上説明したように本発明は、タイムスタンプの考え方を導入することにより、操作権の明示的な取得および解放を要することなく、複数のクライアントプロセス内のデータ(共有データのコピーのデータ)に矛盾が生じないことを保証することができ、しかもその非矛盾性の保証のために通信量を増大させることがないという効果がある。このような効果により、本発明の共有データ制御方式は、電子会議システム等への応用において有効性を発揮する。

【0051】以下に、上述の効果について詳細に説明する。

【0052】① タイムスタンプの管理をサーバプロセスによって統一に行うこと(タイムスタンプを増加することができるのはサーバプロセスのみであり、同時に複数のクライアントプロセスからサーバプロセスに変更要求が送信されたときにはサーバプロセスによるタイムスタンプのチェックにより排他制御が行われる)により、共有データや共有データのコピーのデータに矛盾が生じるような変更要求を確実に拒否することができる。

【0053】② クライアントプロセスは変更要求をまずサーバプロセスに通知してその返事の判定に基づいて実際のデータの変更を行うか否かを決定しているので、クライアントプロセスとサーバプロセスとの間の通信量は変更要求の通信とその返事の通信との2回(1往復)でありタイムスタンプを使用しない場合の通信量と変わらない。すなわち、タイムスタンプの考え方の導入による通信量の増大は生じない。

【0054】③ 従来の共有データ制御方式(操作権方式)におけるような操作権の明示的な取得および解放を行う必要がなくなり、「あるクライアントプロセスにいつまで操作権を保持させるべきか」という困難な決定を行うことが不要になる。

【図面の簡単な説明】

【図1】本発明の一実施例の構成を示すブロック図である。

【図2】図1中のクライアントプロセスがサーバプロセスに変更要求を発行する際の処理を示す流れ図である。

【図3】図1中のクライアントプロセスがサーバプロセスからの変更通知を受け付ける際の処理を示す流れ図である。

【図4】図1中のサーバプロセスがクライアントプロセスからの変更要求を受け付ける際の処理を示す流れ図である。

【図5】図1に示す共有データ制御方式の具体的な動作を説明するためのブロック図である。

【符号の説明】

1〜3 クライアントプロセス

4 サーバプロセス

50 5〜7 プロセス間通信

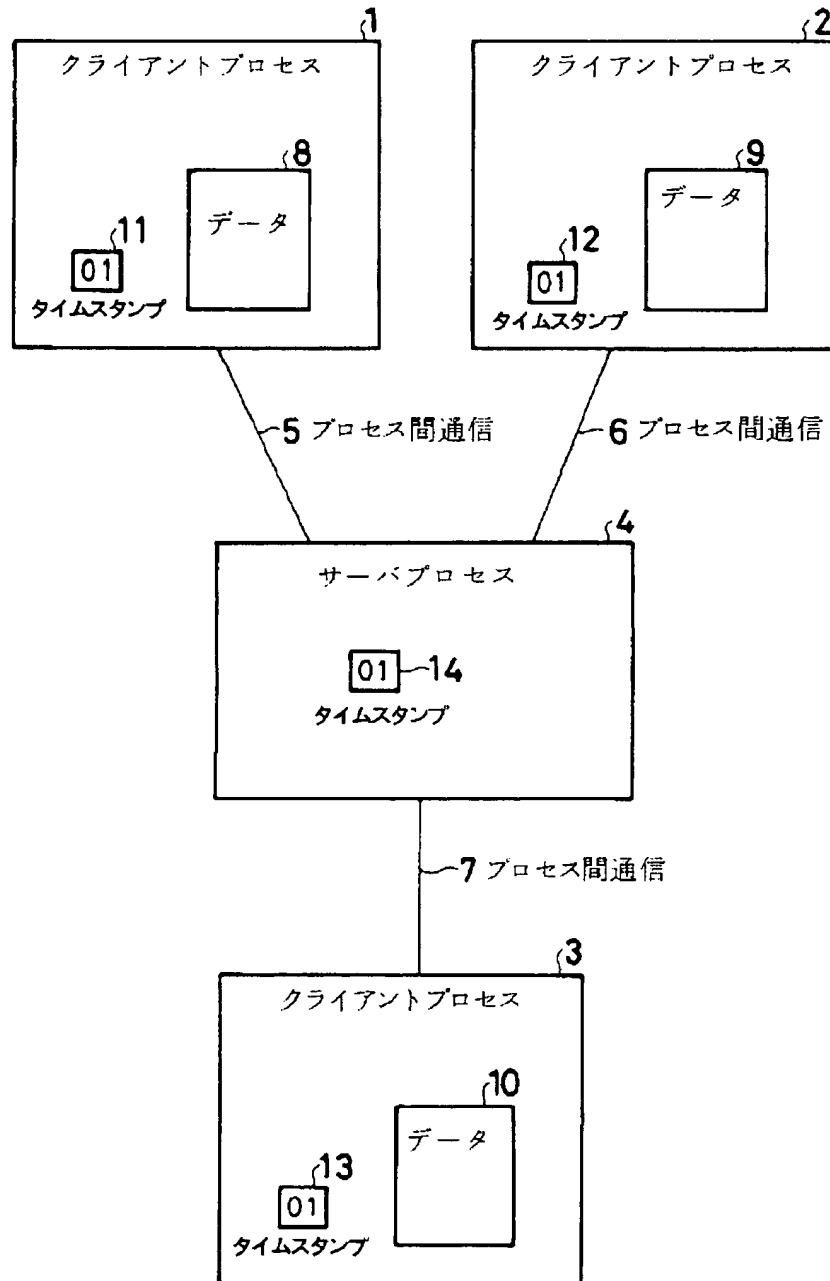
8~10 データ

51 変更通知

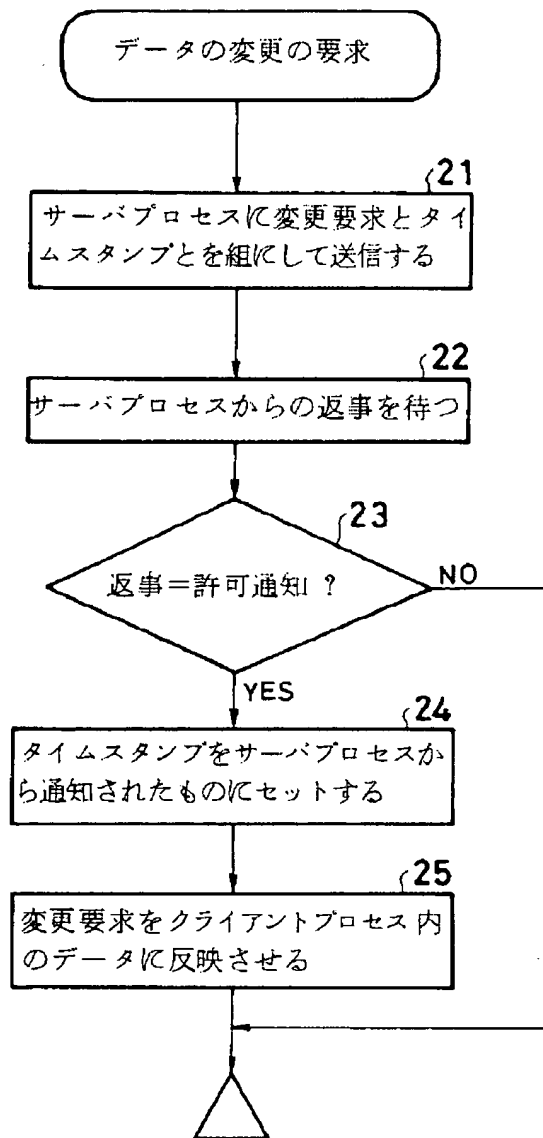
11~14 タイムスタンプ

52, 53 変更要求

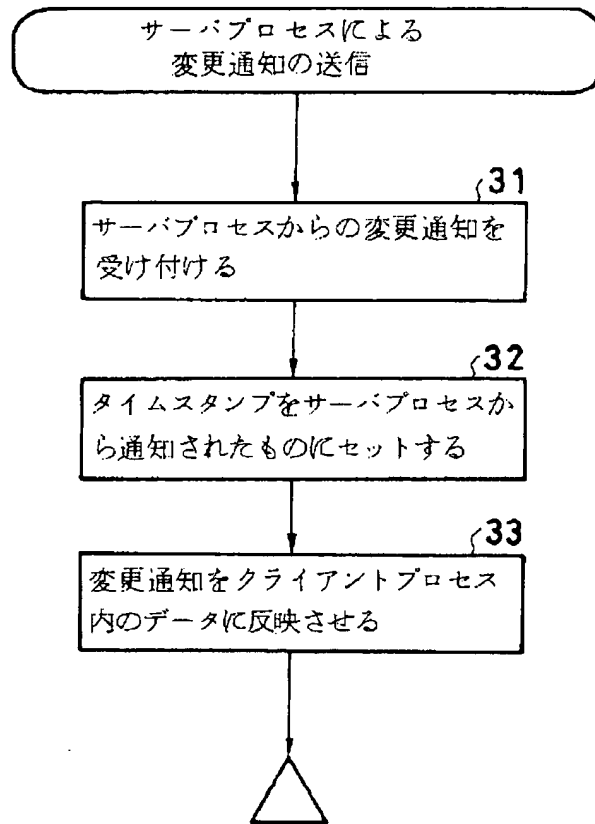
【図1】



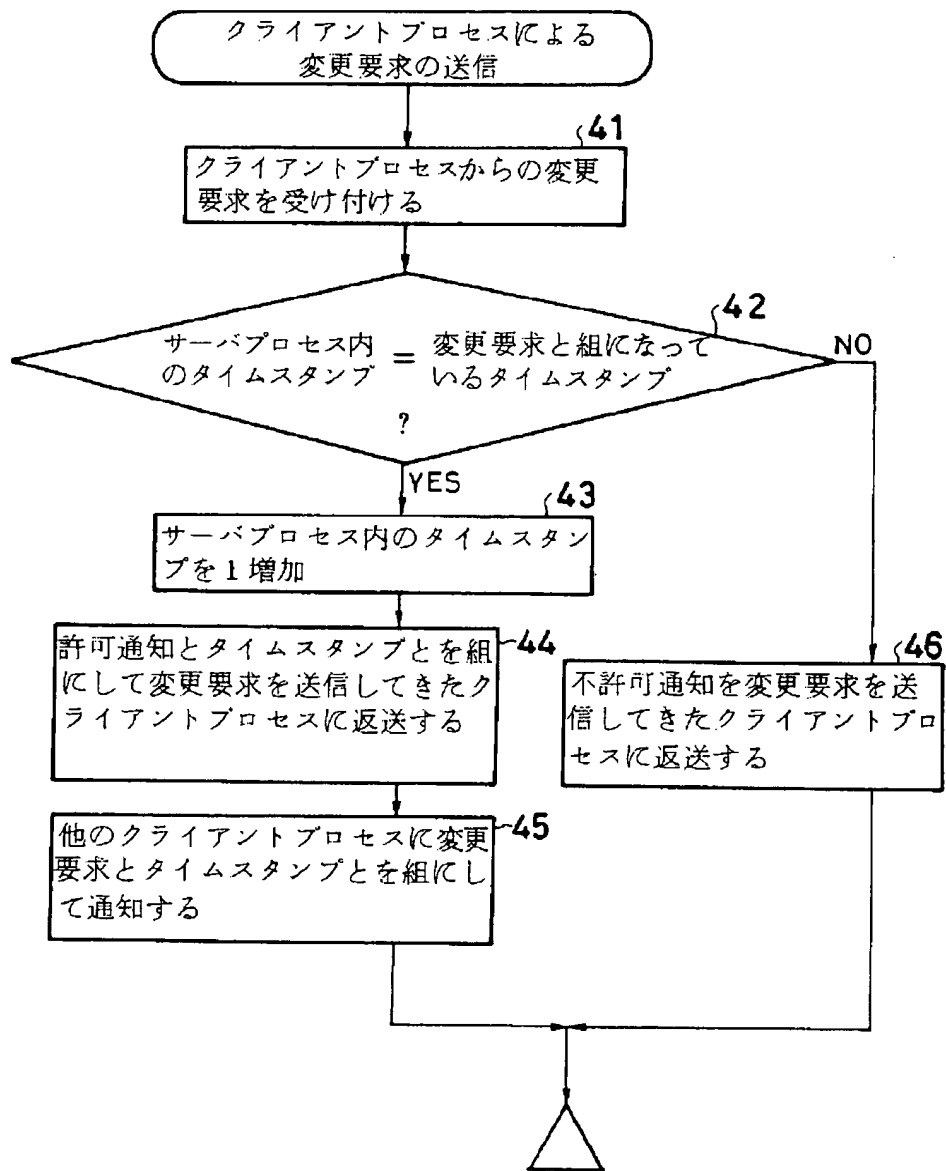
【図2】



【図3】



【図4】



【図5】

